

# 基于STM32的 PMSM FOC软件库 培训




MCU Application  
Great China

蒋建国

Shanghai,  
March, 2008



# Agenda



## 第一天

### 上午

- 培训人员介绍
- STM32 产品技术介绍
- 马达控制套件

### 下午

- 直流无刷马达介绍
- FOC 基础
- Clark及Parke坐标系转换
- Circle limitation
- 磁链及力矩控制器
- 电流读取

# Agenda

## 第二天

### 上午

#### 转子的速度/位置反馈

- Hall传感器
- 正交编码器

#### 无传感器转子位置检测

- 框图
- 观测器理论
- 应用到PMSM马达系统

### 下午

#### 无传感器转子位置检测

- PLL
- 起动策略
- 转子堵转检测
- 头文件 “MC\_State\_observer\_param.h”

# Agenda

## 第三天

### 上午

- 软件库结构及其它特性
- 无传感器模式软件开发进程；
- 头文件：MC\_Control\_param.h；
- 怎样计算转矩/磁链PI参数的初始值；
- 怎样计算观测器增益参数的初始值

### 下午

- 练习

# Plan



## ▣ 培训人员介绍

## ▣ STM32 产品技术介绍

### ▣ Cortex-M3简介

#### ▣ 内核

#### ▣ NVIC

#### ▣ PWM 产生

#### ▣ 速度/位置 反馈

#### ▣ 多定时器配置

#### ▣ ADC

## ▣ 马达控制套件



# Cortex-M3 处理器



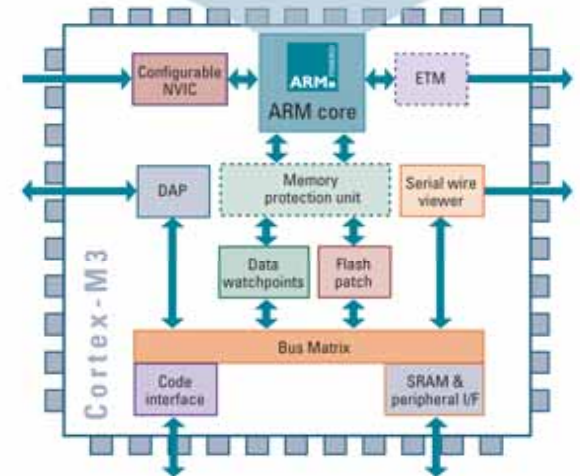
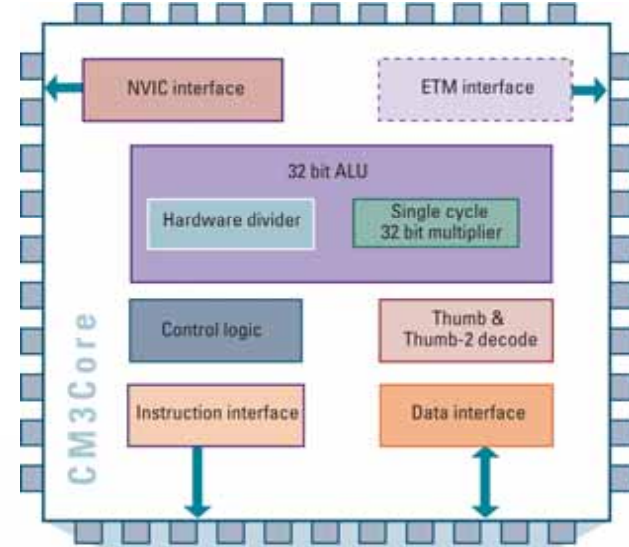
处理器集成了先进的内核及系统外设

## STM32 Cortex-M3 内核



- 哈佛结构
- 3级流水线 & 分支指令推断
- Thumb<sup>®</sup>-2 指令
- ALU & 硬件除法 & 单周期乘法

## Cortex-M3 处理器

- Cortex-M3 内核
- 可配置的中断控制器
- 总线矩阵
- 先进的debug 模块
- 可选的 MPU & ETM (Not available in STM32F10x)



# Cortex-M3处理器概述 (1/2)

- 
- 
- ARM v7M 结构
  - Thumb-2 指令组
    - 混合16及32位指令，具有非常高的代码密度
  - 哈佛结构
    - 独立的指令和数据总线允许并行进行指令读取及数据存储
  - 集成可嵌套的向量式中断控制器(NVIC)，实现中断的快速响应
  - 中断向量表内存放ISR的地址而非指令(与ARM7不同)
  - 使完全用C语言编程成为可能
    - 即便是：复位，中断和异常
  - 集成总线矩阵
    - 总线仲裁
    - 位域 – 原子位操作
    - 写缓冲
    - 存储器接口(指令 & 数据) 及 系统接口 & 外设总线
  - 集成系统定时器 (SysTick)，可用于实时OS或任务调度

# Cortex-M3处理器概述(2/2)



## 3级流水线

- 读指令, 解码 & 执行
- 单周期乘法
- 单周期乘累加

Source	Destination	Cycles
16b x 16b	32b	1
32b x 16b	32b	1
32b x 32b	32b	1
32b x 32b	64b	3-7*

\*UMULL, SMULL, UMLAL, and SMLAL 可被中断, 其依赖于source的值

## 硬件除法

- UDIV & SDIV (Unsigned or Signed divide)
- 指令执行时间: 2到12 周期 (具体由除数和被除数决定)
- 除数和被除数越接近, 执行时间越短
- 可被中断(丢弃/重启)





# 异常/中断处理



- ❑ 非常低的中断响应时间
  - ❑ 异常工作在特权模式下
  - ❑ LDM/STM 指令可被中断，以实现低中断响应时间
  - ❑ 处理器状态自动保存和恢复
    - ❑ 大大减少 ISR 进入和退出时间
    - ❑ 允许 ISR 的 'C'编程
- ❑ Cortex-M3 处理器集成可嵌套的向量式中断控制器(Nested Vectored Interrupt Controller---NVIC)
  - ❑ 43 个可屏蔽中断通道(不包括16个Cortex-M3自带的中断通道)
  - ❑ 16 个可编程优先级
  - ❑ 允许中断的早期处理
  - ❑ 先进的中断响应机制
    - ❑ 末尾连锁 - Tail-chaining
    - ❑ 迟来 - Late-arrival



# 中断响应机制



- 中断响应无需指令操作

- 进入

- 处理器状态自动通过数据总线保存到堆栈中

- {PC, xPSR, R0-R3, R12, LR}

- 同时，ISR预取在指令总线上。

- 一旦堆栈PUSH结束，ISR可直接执行。

- 迟来中断须重新ISR预取，但处理器状态保存不需重复。

- 退出

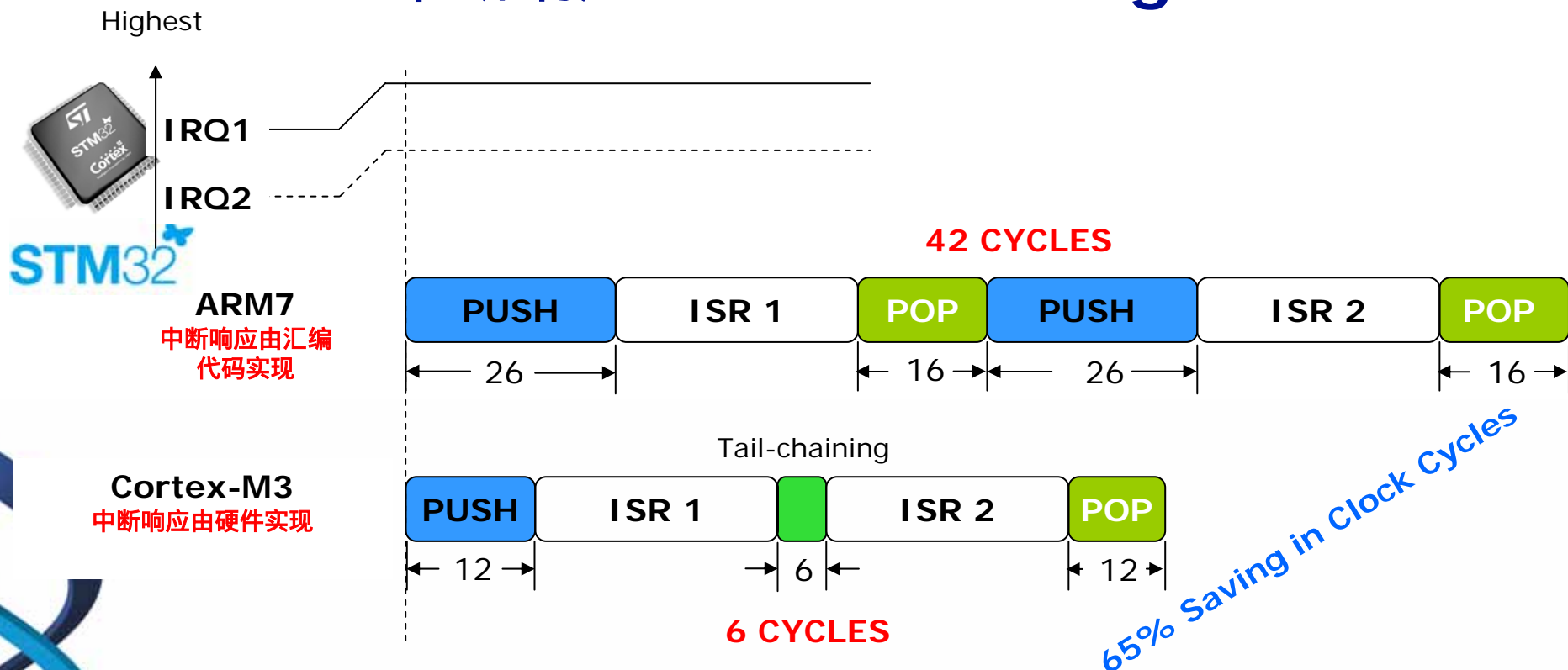
- 处理器状态自动从堆栈恢复

- 同时，被中断的指令被预取，一旦堆栈POP结束，其马上执行

- 堆栈POP可被中断，允许新的ISR马上执行，不需要处理器状态保存到堆栈中。



# 中断响应 - Tail Chaining



## ARM7

- 从IRQ1请求到进入ISR1：26个周期
  - 若中断LSM指令：42个周期
- 从ISR1退出到ISR2进入：42个周期
- 从ISR2返回：16个周期

## Cortex-M3

- 从IRQ1请求到进入ISR1：12个周期
  - 若中断LSM指令：12个周期
- 从ISR1退出到ISR2进入：6个周期
- 从ISR2返回：12个周期

# NVIC 寄存器

中断输入的每个通道由几个寄存器控制，包括：

## 使能/禁止位

- 使能/禁止中断

- 该控制位可置1，清0或被读

## 中断请求位

- 可通过令中断请求位置1来实现中断请求

- 一个中断请求只有在其被使能且优先级最高时才响应

- 该控制位可置1，清0或被读

## 中断激活位--Active Bit

- 置1：表示中断正在执行或被另一个更高优先级的中断嵌套（“active-stacked”）

- 只读

## 优先级域

- 每个中断有4个优先级定义位

# 中断优先级

- 每个中断有4个优先级位
- 4个优先级位分为抢占式优先级位及非抢占式子优先级位
  - 非抢占式子优先级位只有在抢占式优先级相同时才起作用
  - NVIC寄存器的PRIGROUP域定义了抢占式优先级位 (“group-priority”) 的位数和非抢占式子优先级位 (“sub-priority”) 的位数
  - Group priority 定义抢占式优先级
- 数值越低则优先级越高
- 中断序号越低则优先级越高
  - 例如：如果优先级寄存器值相同，IRQ3优先级高于IRQ4

PRIGROUP (3 Bits)	Binary Point (group.sub)		Preempting Priority (Group Priority)		Sub-Priority	
			Bits	Levels	Bits	Levels
011	4.0	gggg	4	16	0	0
100	3.1	gggs	3	8	1	2
101	2.2	ggss	2	4	2	4
110	1.3	gsss	1	2	3	8
111	0.4	ssss	0	0	4	16

# Cortex-M3 中断类型

No.	Exception Type	Priority	Type of Priority	Descriptions
1	Reset	-3 (Highest)	fixed	Reset
2	NMI	-2	fixed	Non-Maskable Interrupt
3	Hard Fault	-1	fixed	Default fault if other handler not implemented
4	MemManage Fault	0	settable	MPU violation or access to illegal locations
5	Bus Fault	1	settable	Fault if AHB interface receives error
6	Usage Fault	2	settable	Exceptions due to program errors
7-10	Reserved	N.A.	N.A.	
11	SVCALL	3	settable	System Service call
12	Debug Monitor	4	settable	Break points, watch points, external debug
13	Reserved	N.A.	N.A.	
14	PendSV	5	settable	Pendable request for System Device
15	SYSTICK	6	settable	System Tick Timer
16	Interrupt #0	7	settable	External Interrupt #0
.....	<b>In STM32F10x 43 Interrupts are implemented (total interrupts available 59)</b>			
256	Interrupt#240	247	settable	External Interrupt #240

# Plan



STM32

## ▣ Cortex-M3 简介

▣ 内核

▣ NVIC

▣ PWM 产生

▣ 速度/位置 反馈

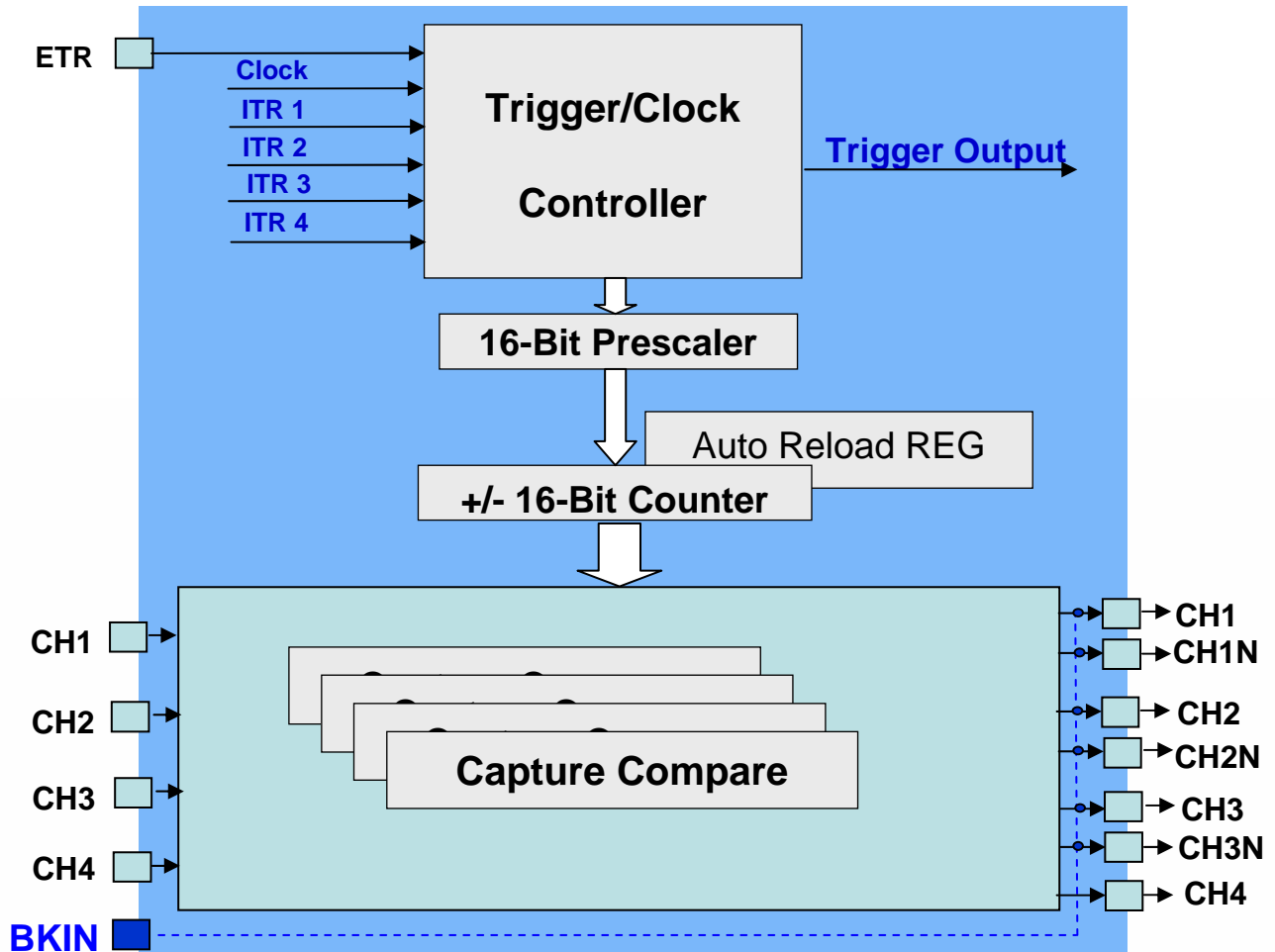
▣ 多定时器配置

▣ ADC

▣ 马达控制套件



# 高级定时器





# 高精度 PWM 产生



## ▣ 时钟

- ▣ 可为APB总线频率的2倍
- ▣ 最大72MHz：可提供13.8ns 定时精度

## ▣ 边沿或中心对称模式

## ▣ 更新率倍频模式(见下页)

- ▣ 中心对称模式下无精度损失
- ▣ 每个PWM周期可产生两次中断或DMA

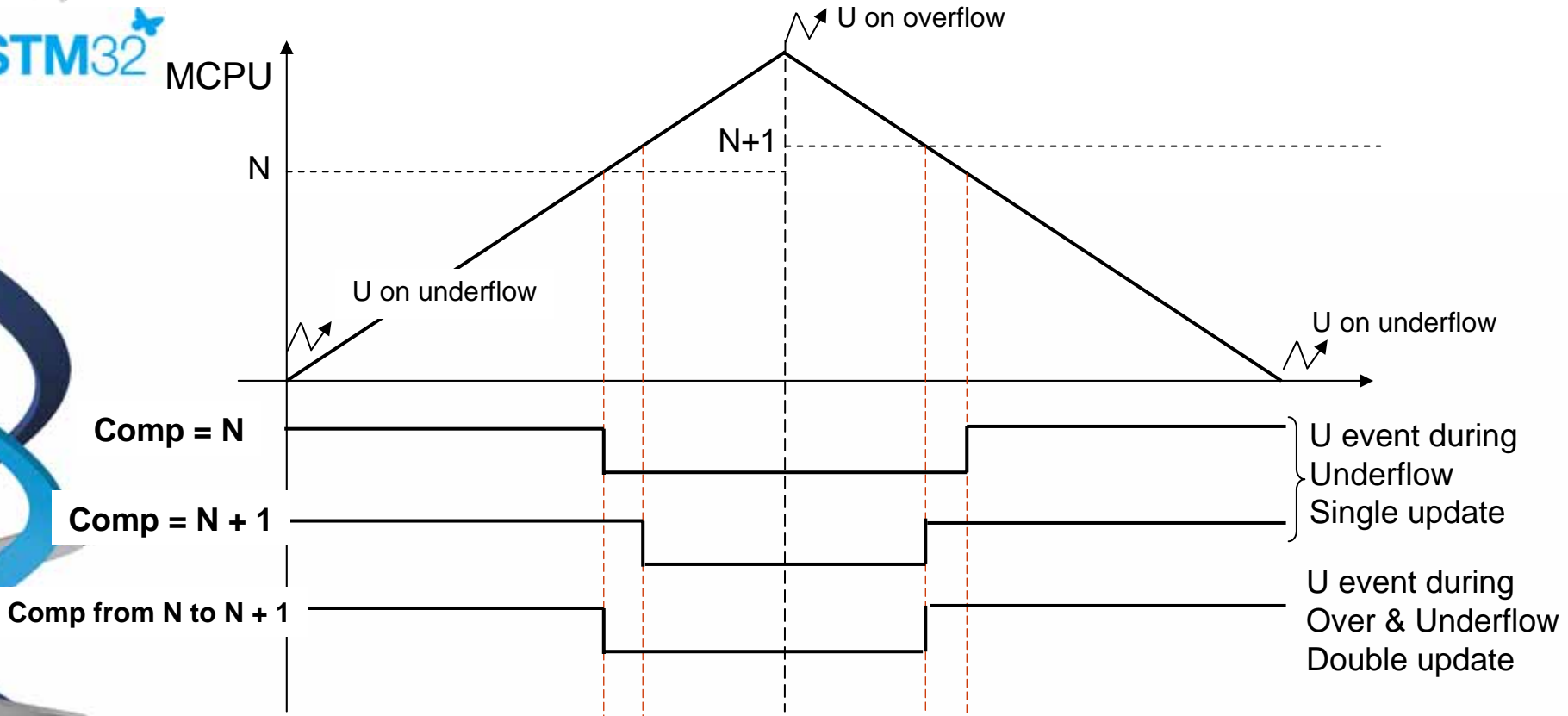


# 更新率倍频模式

在PWM计数器上溢时产生更新事件 (U)可提高 duty cycle的精度



STM32



# PWM 定时器的主要中断

## U (Update) 事件

### 同步更新所有预载寄存器，包括：

#### 4个比较寄存器（设定duty cycle）

##### 预载机制可禁止

#### 1个PWM周期寄存器

##### 允许在定时器工作时改变PWM周期而duty cycles不变

#### PWM时钟预分频器

## 调整U事件速率

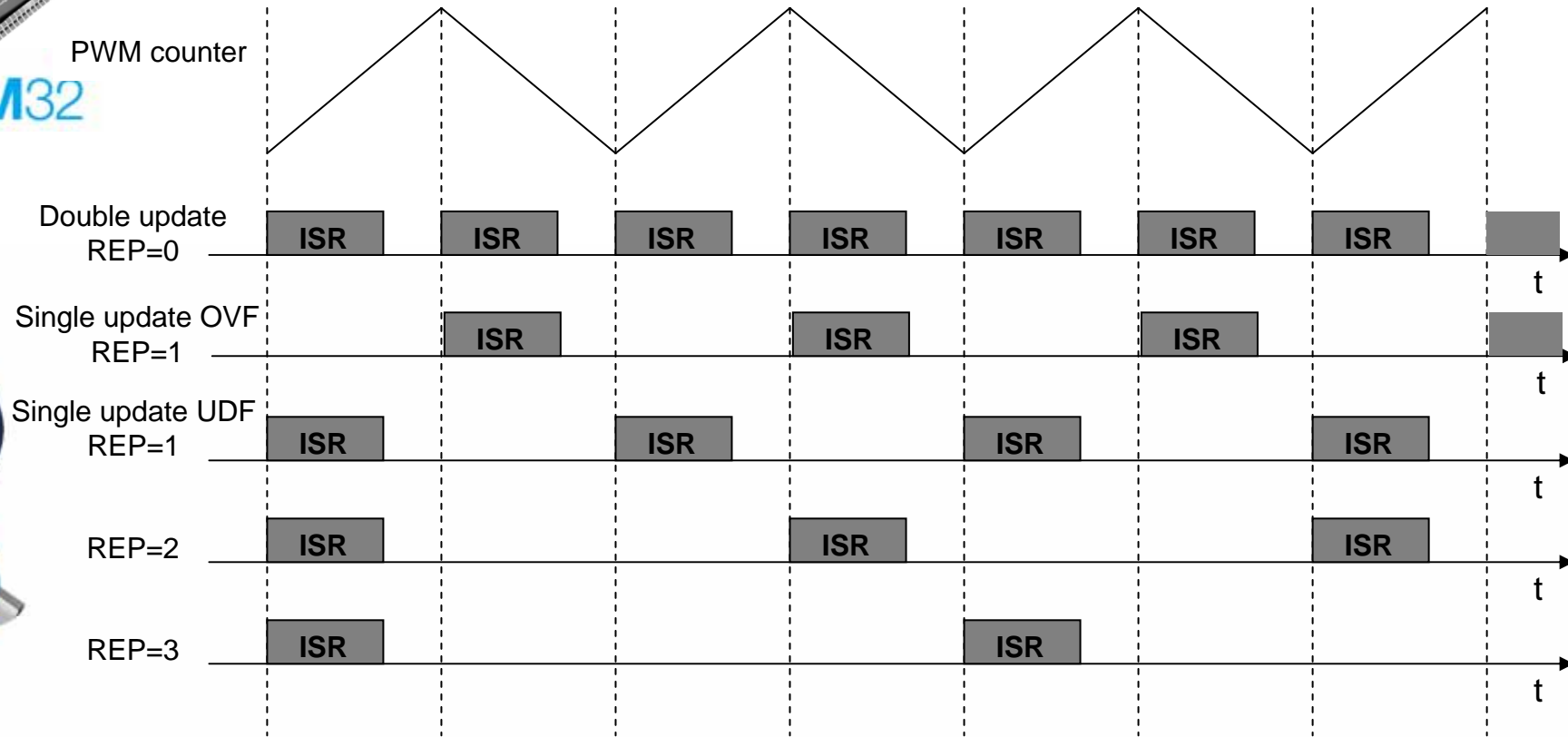
### 通过编程8位重复计数器实现

### 允许选择定时器上溢/下溢或两者中的任一个用于重复计数器计数（由计数方式决定）

# 重复计数器



STM32

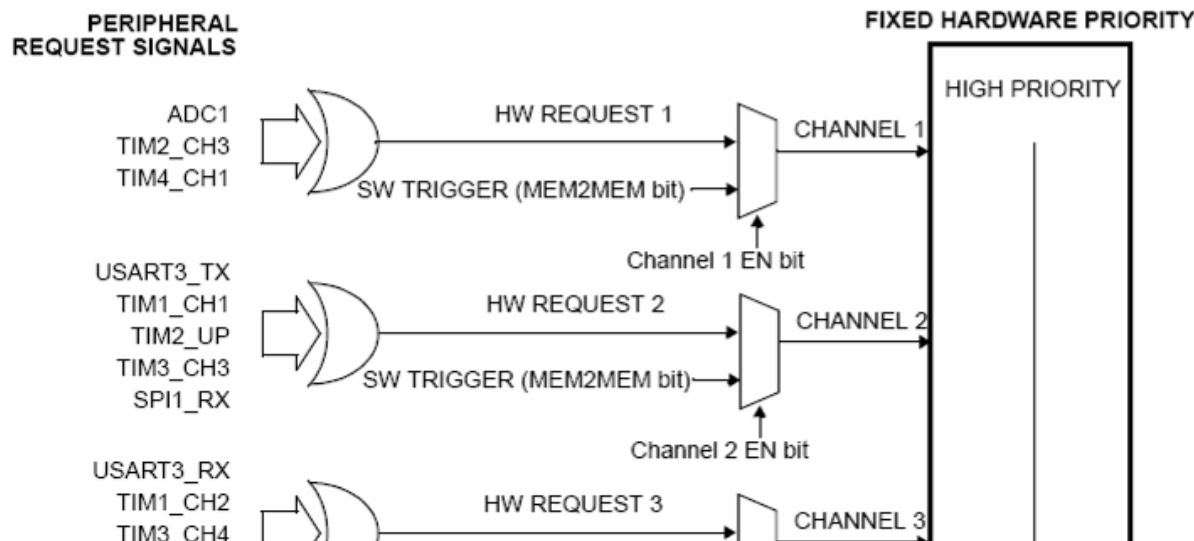


# 其它中断及DMA

## ❏ PWM定时器的其它中断

- ❏ 输出比较匹配(向上/向下计数方向可选)/输入捕捉
- ❏ 触发事件
- ❏ 故障停机

## ❏ 有些事件还映射在DMA控制器上



# PWM定时器 DMA的连续传输 (Burst mode)

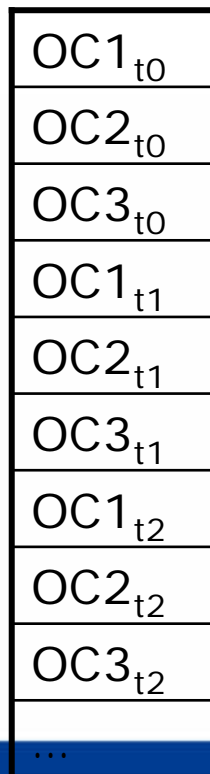


允许用一个DMA事件更新几个寄存器

有效地使用DMA (仅要求一个数据流)

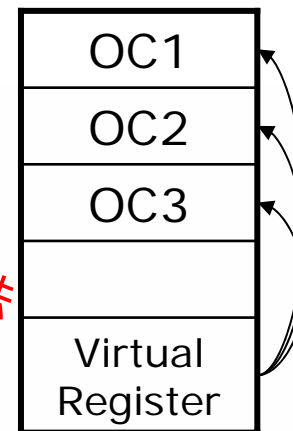
STM32

RAM



目标

Registers



实现方法



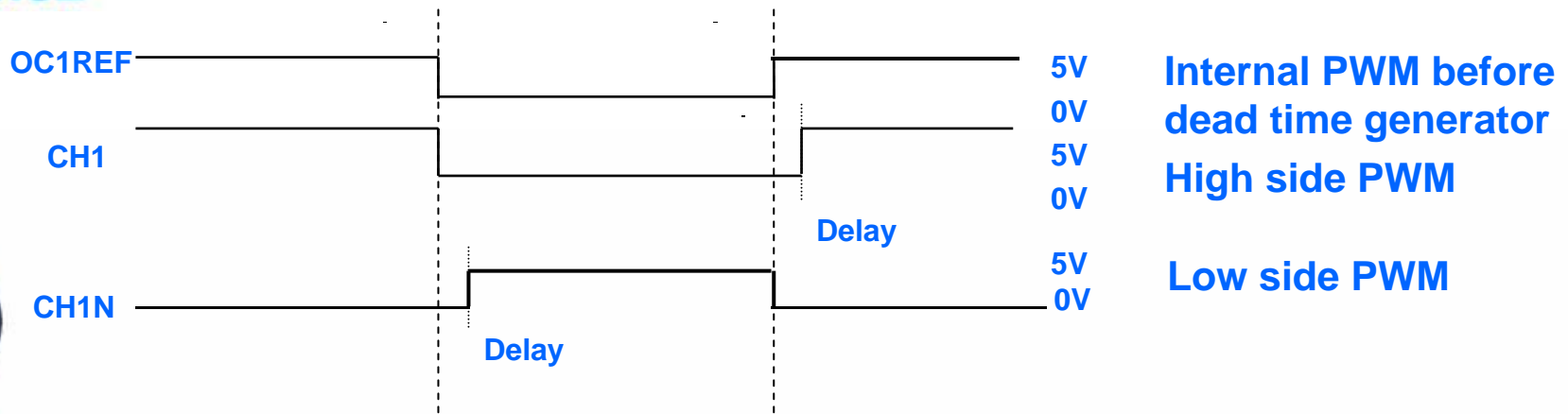
# PWM 输出管理



STM32

## 可编程的死区产生

- 8位寄存器，在时钟为72MHz时13.8ns 最大精度(from 0 to 14 $\mu$ s, 非线性)



- 各自极性可编程
- 专门的故障停机输入
  - 关闭6个PWM输出且发出中断请求
  - 异步操作**(无须时钟同步)

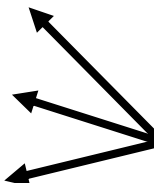

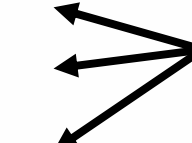





# 灵活的PWM输出 1/2



STM32

Control bits					Output states <sup>(1)</sup>	
MOE bit	OSSI bit	OSSR bit	CCxE bit	CCxNE bit	OCx output state	OCxN output state
1	X	0	0	0	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
					Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
					OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
					OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
					Output Disabled (not driven by the timer) OCx=CCxP, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=CCxNP, OCxN_EN=0
					Off-State (output enabled with inactive state) OCx=CCxP, OCx_EN=1	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
					OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state) OCxN=CCxNP, OCxN_EN=1
					OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
0	X	0	0	0	Output Disabled (not driven by the timer)	
					Asynchronously: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0	
					Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state.	
					Off-State (output enabled with inactive state)	
					Asynchronously: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1	
					Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state	

- 
 PWM timer used as a GP timer
- 
 Motor Control (sinewave)
- 
 Motor Control (6-steps)
- 
 Motor Control (sinewave)
- 
 Outputs disconnected from I/O ports
- 
 All PWMs OFF (low Z for safe stop)



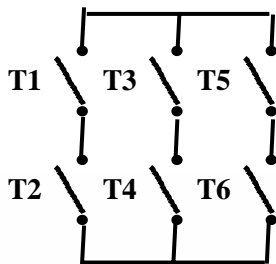


# 灵活的PWM输出 2/2

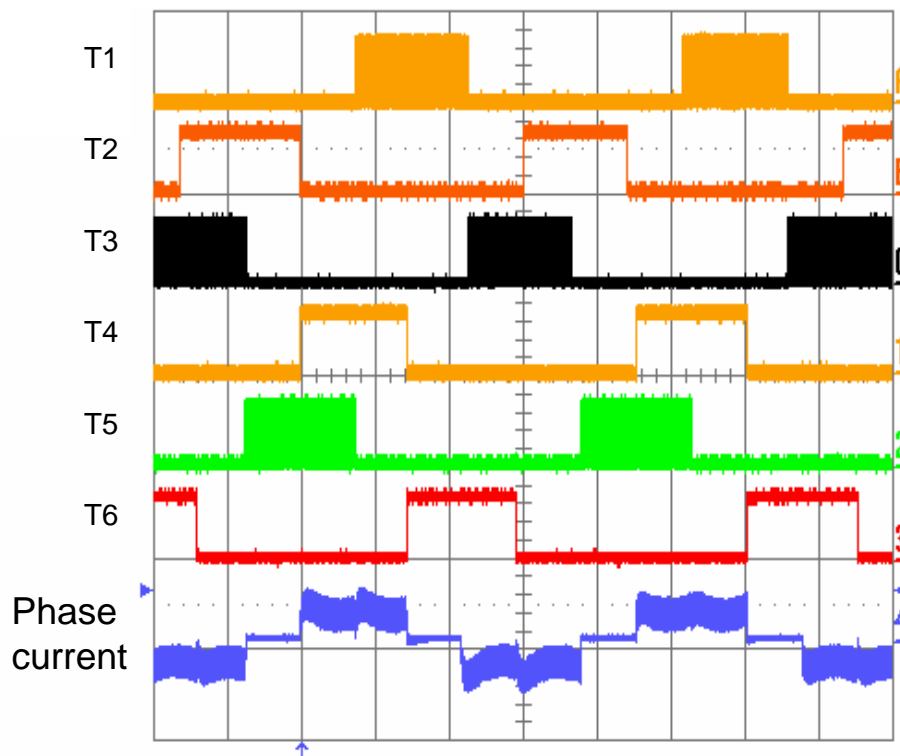
## 例子: BLDC 6-steps 驱动



STM32



Step	High	Low	OC1	OC1N	OC2	OC2N	OC3	OC3N
1	T1	T4	oc1ref	0	0	1	0	0
2	T1	T6	oc1ref	0	0	0	0	1
3	T3	T6	0	0	oc2ref	0	0	1
4	T3	T2	0	1	oc2ref	0	0	0
5	T5	T2	0	1	0	0	oc3ref	0
6	T5	T4	0	0	0	0	oc3ref	0



# 故障停机输入



## 故障停机事件产生:

- BRK引脚输入，其极性可编程且可使能
- 时钟安全系统

STM32

## 当故障停机发生时:

- MOE (输出主控位) = 0
- 故障停机标志置1，可用于产生中断
- 每个输出通道的电平由OISx 位决定
  - 例如对PMSM马达，所有下桥臂打开可用于刹车

## 故障停机事件应用:

- 若AOE=0，MOE位保持0直到软件重新对其写1
  - BRK引脚连接功率模块的故障反馈信号，一般用于保护功率器件
- 若AOE=1，MOE位在下次U事件时自动置1
  - 一般用于实现电流调节（调节频率=U事件频率）
  - 电流调节还可用外部触发输入实现(ETR)



# “禁烟”保护

■ 关键寄存器可被上锁，用于保护功率器件(如软件跑飞，...)，如：

■ 死区时间设定，PWM输出极性，故障输入使能，...

■ 所有寄存器在上锁前可读写（上锁后只读）

■ 一旦上锁位被写，其值不可改变，除非MCU复位(即：此位只可写一次)

■ 4个上锁等级提供了非常高的灵活性，可根据不同应用设置不同等级

■ GPIO的配置也可被上锁，避免PWM输出口被设置为标准输出口

# Debug 特性



❑ 马达控制类应用对于设置断点debug需慎重对待

❑ 普通的断点可能导致功率器件损坏

❑ 闭环系统在断点停止后无法在断点处继续执行

STM32

❑ 专门有一个控制位允许配置在断点发生时PWM定时器的行为

❑ 普通模式: 定时器继续运行

❑ 由于此时固定的duty cycle加到功率模块上, 有些情况下非常危险

❑ 安全模式: 定时器停止运行, PWM输出被关断

❑ 保护功率模块, 且定时器在断点处继续运行



# Plan



STM32

## ▣ Cortex-M3 简介

▣ 内核

▣ NVIC

▣ PWM 产生

▣ 速度/位置 反馈

▣ 多定时器配置

▣ ADC

▣ 马达控制套件



# 速度反馈

- 由通用定时器使用专门的模式处理

  - 适用于所有的定时器

- Hall传感器

  - Hall 接口 (内部对三个Hall输入进行异或)

- 正交编码器

  - 编码器模式1, 2 & 3 (2x, 4x)

- 转速传感器

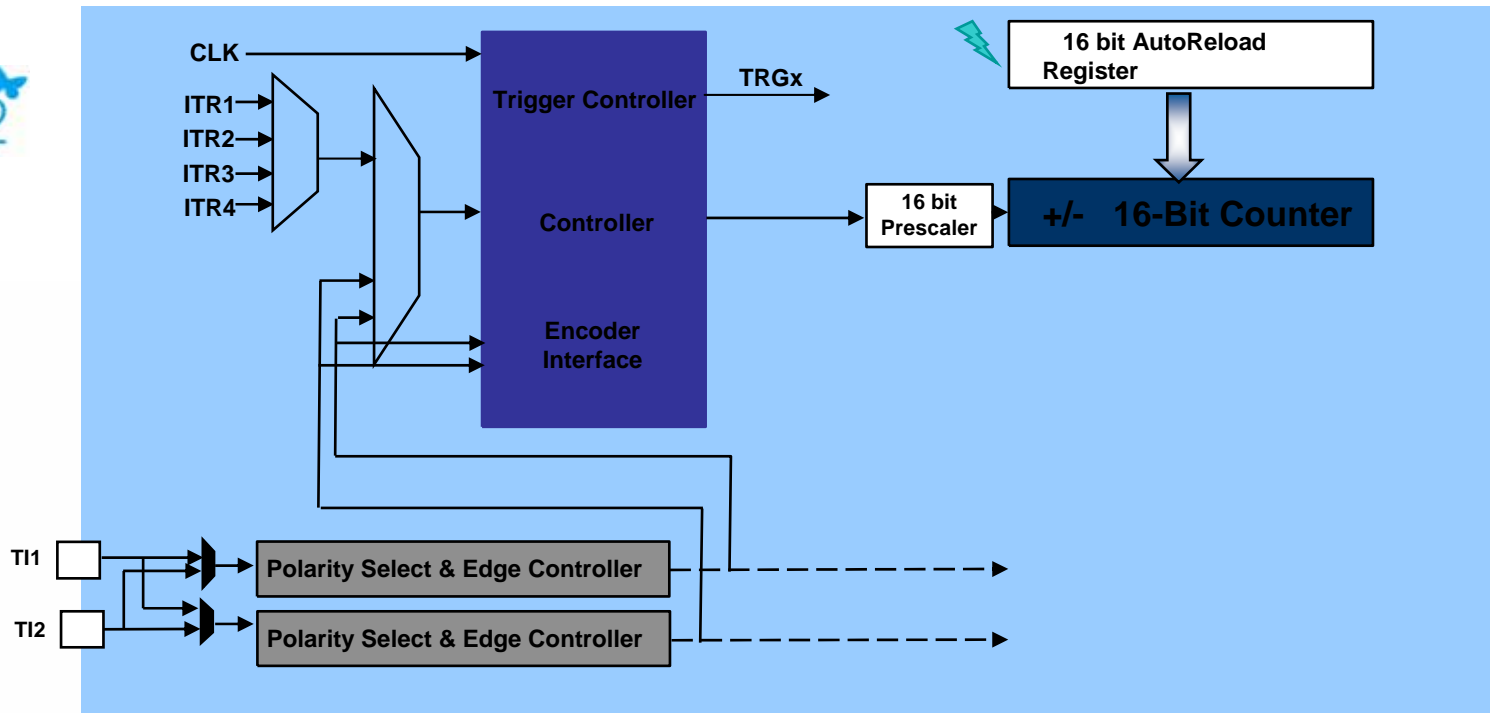
  - 输入捕捉方式, 测量周期 (在捕捉的同时清计数器)



STM32



# TIM正交编码器模式框图



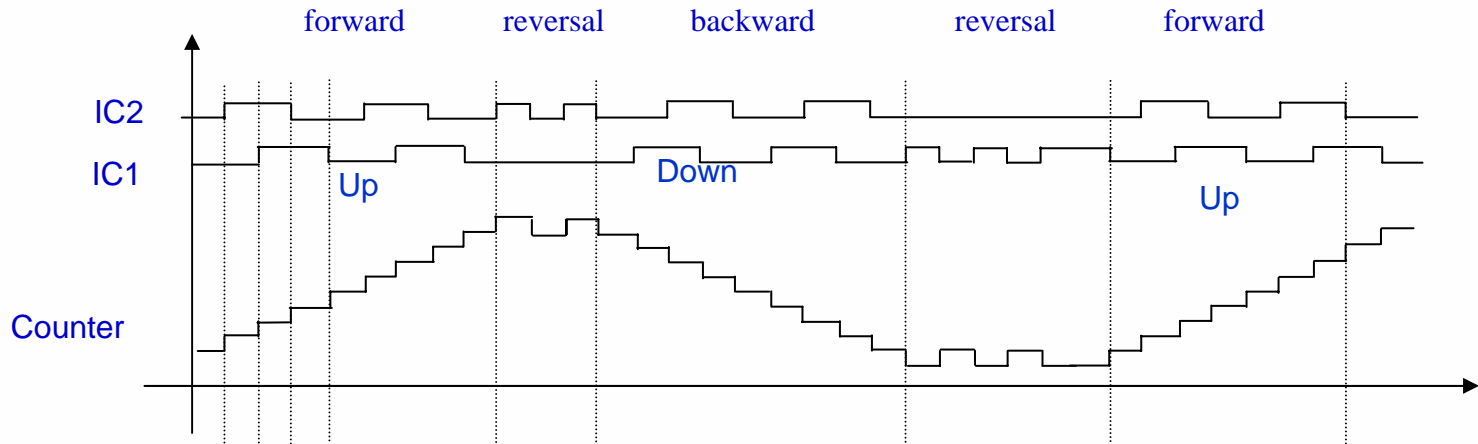
# TIM与正交编码器接口



## 正交编码器与STM32连接举例:

- 增量式正交编码器可直接与MCU相连而无需外部逻辑电路
- 正交编码器的第三个输出 (Z : 机械零位), 可连至外部中断口来触发定时器的计数器复位

### Example of counter operation in Encoder Interface mode





# 正交编码器关键特性

## 可编程的计数率

- ▣ x4: 标准模式，所有边沿有效
  - ▣ 1000线的正交编码器每转产生4000个计数脉冲
- ▣ x2: 只对A (或B)计数，但仍可确定方向
- ▣ “转速模式”: 正交编码器输入可分频

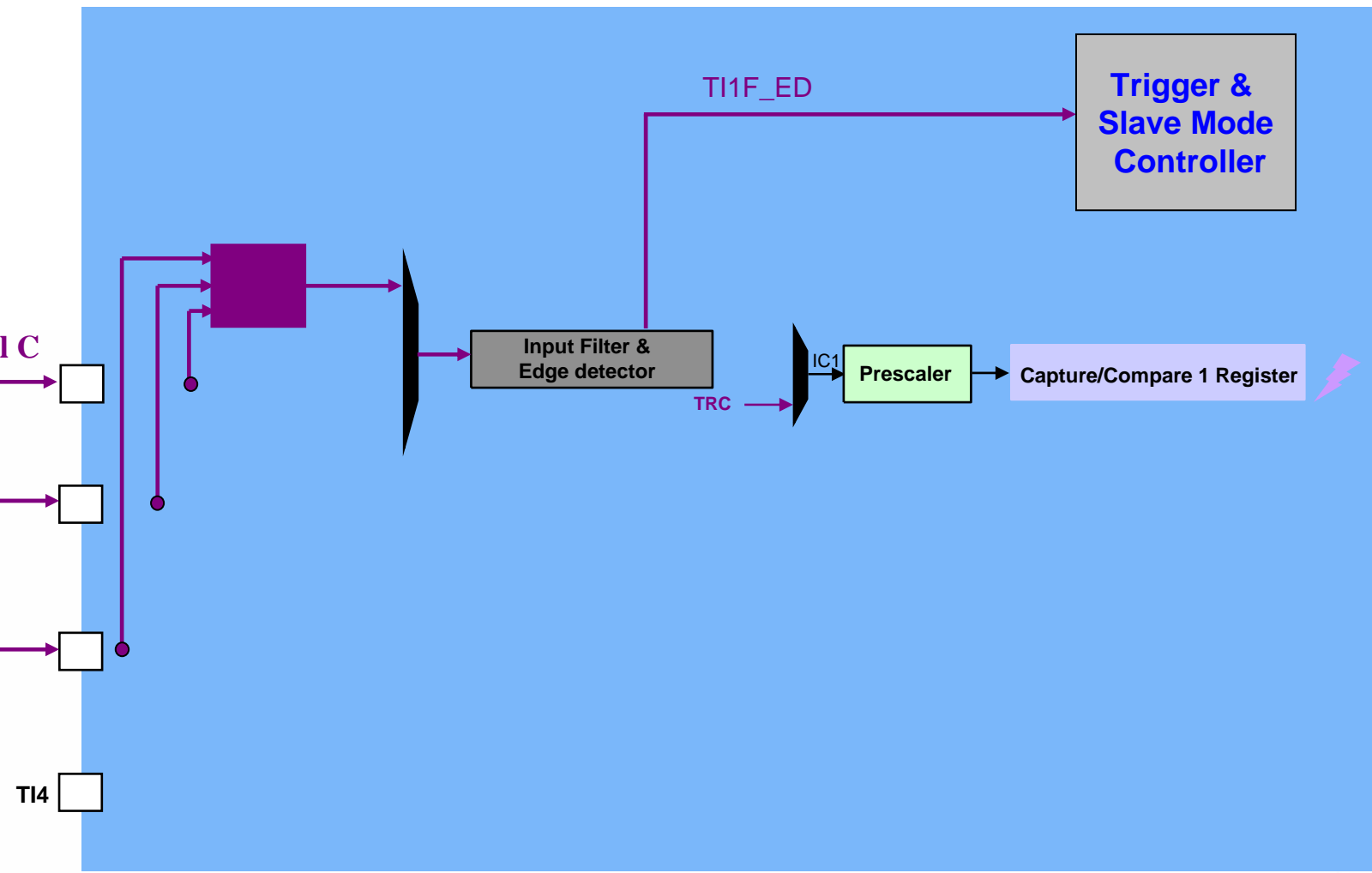
## 精度可编程

- ▣ 当自动重载寄存器的值配置为正交编码器每转产生的计数脉冲时，则计数器的值直接为转子的角度/位置
- ▣ 当自动重载寄存器的值配置为0xFFFF时，与使用自由运行定时器的设计相同

## 编码器每转一周可发出一个或多个中断

- ▣ 一个，每360°；
- ▣ 多个，每60°，90°，...（依赖于自动重载寄存器的配置）

# TIM 与 Hall接口



# Plan



## Cortex-M3 简介

-  内核
-  NVIC
-  PWM 产生
-  速度/位置 反馈
-  多定时器配置
-  ADC
-  马达控制套件

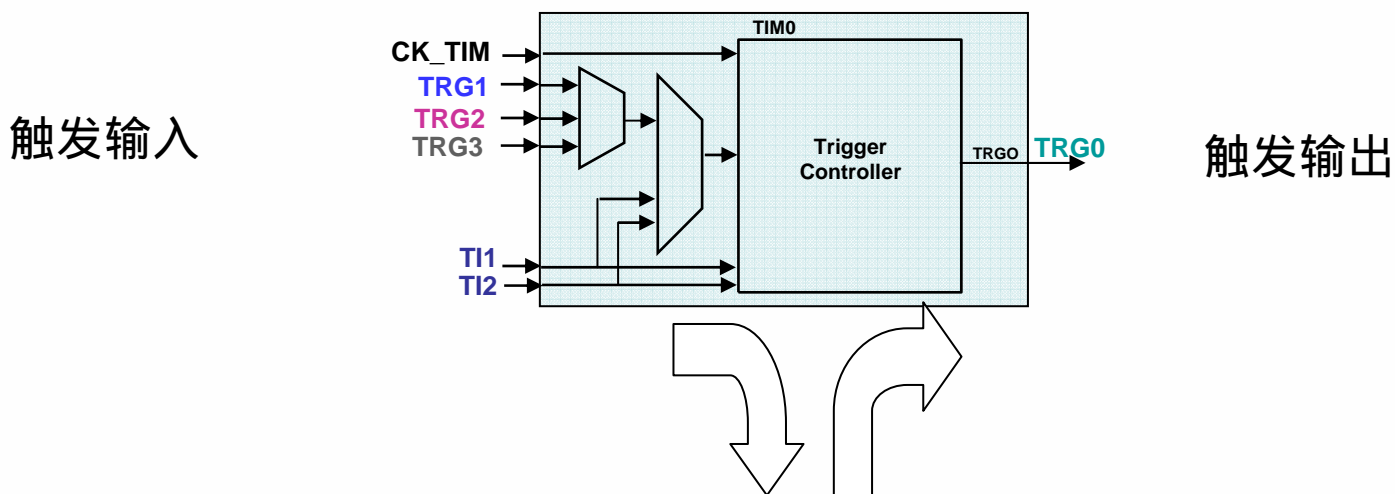


# 定时器同步系统



由于拥有触发输出及多个可选的触发输入，3个通用定时器与PWM定时器能连接在一起串联或同步使用

输入脚TI1 及 TI2也可用作触发器



定时器控制信号: clock, reset, update, enable,...



# 同步模式配置



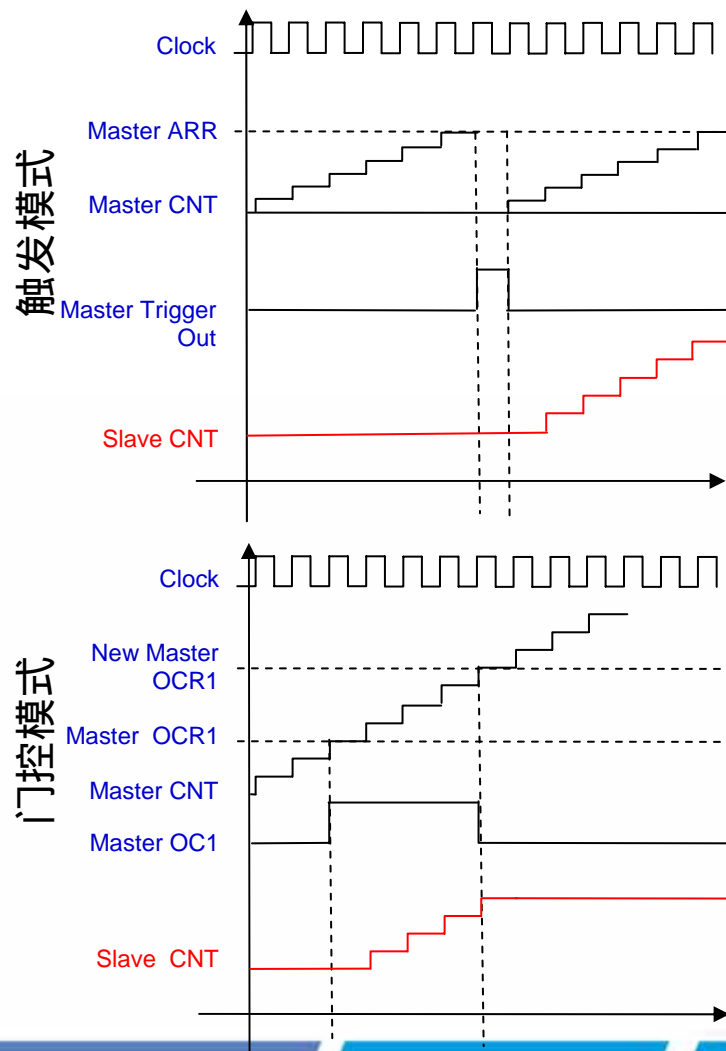
STM32

配置为主模式时，触发器输出可为：

- 计数器复位
- 计数器使能
- 更新事件
- 输出比较信号

当配置为从模式时，定时器可工作在如下模式：

- 触发模式
- 门控模式
- 复位模式
- 外部时钟模式

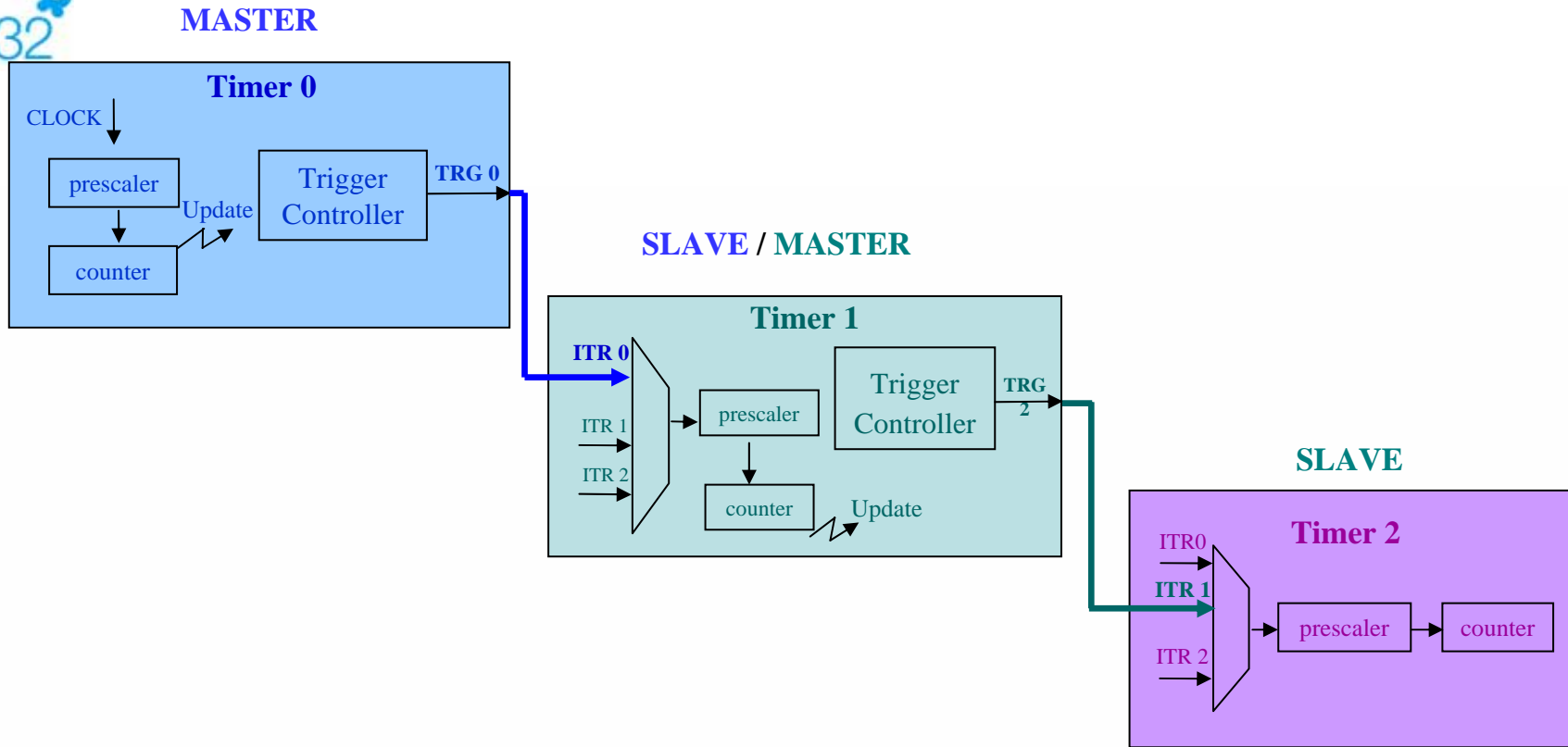


# 例 1/3: 定时器级联

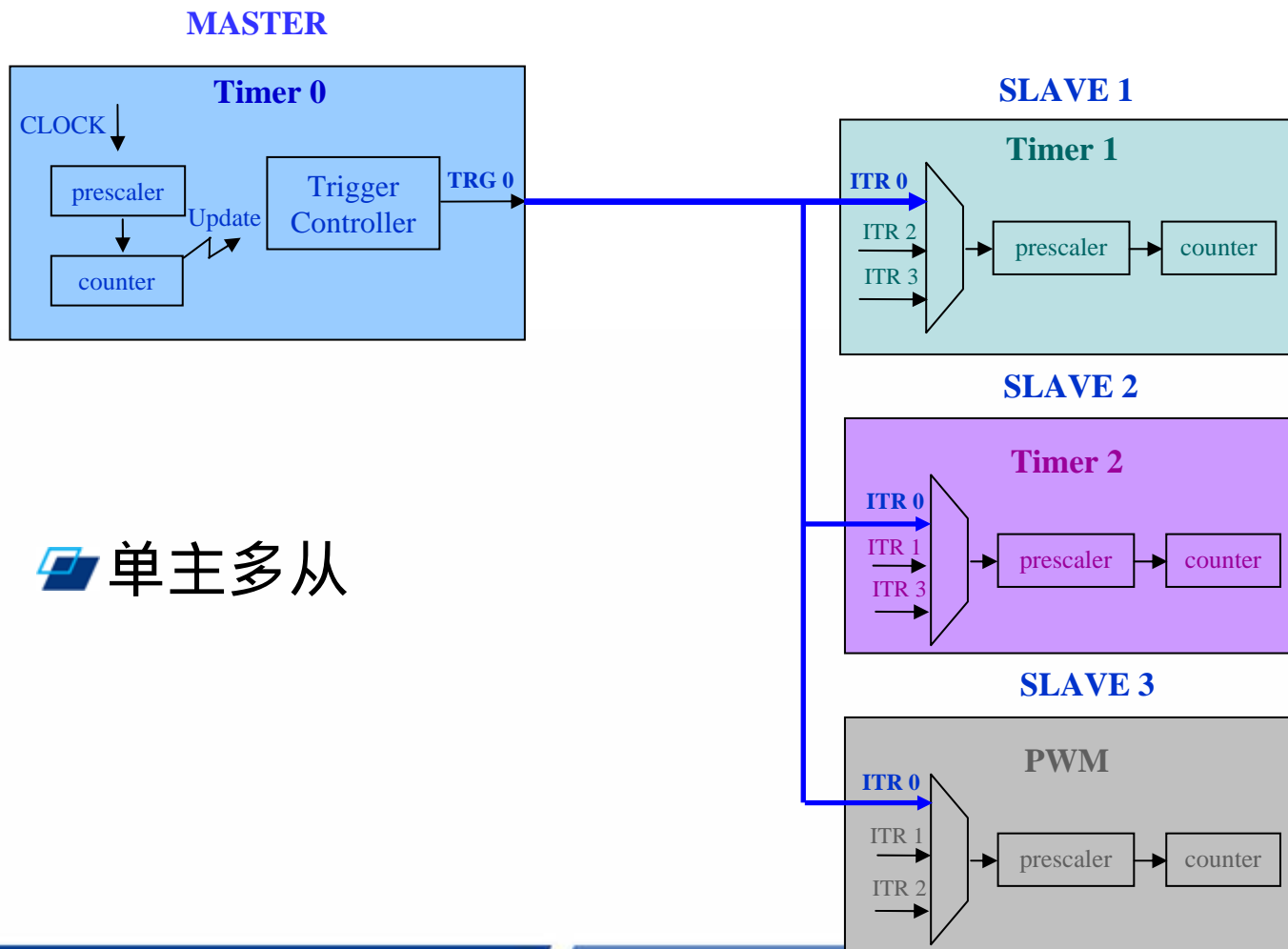
级联模式 (从定时器工作在外部时钟模式)



STM32



# 例 2/3: 同步启动



单主多从



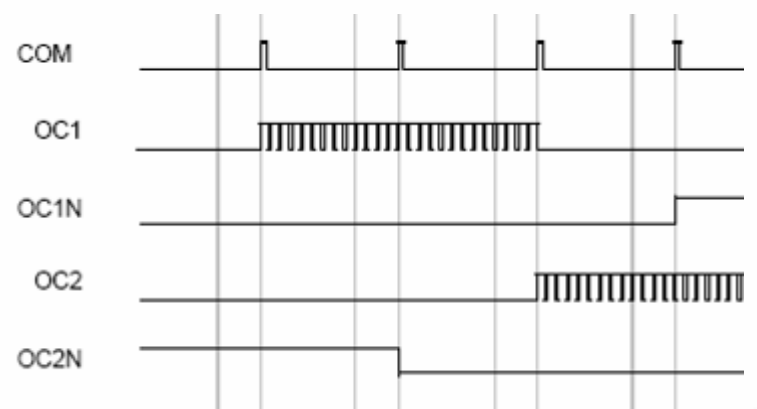
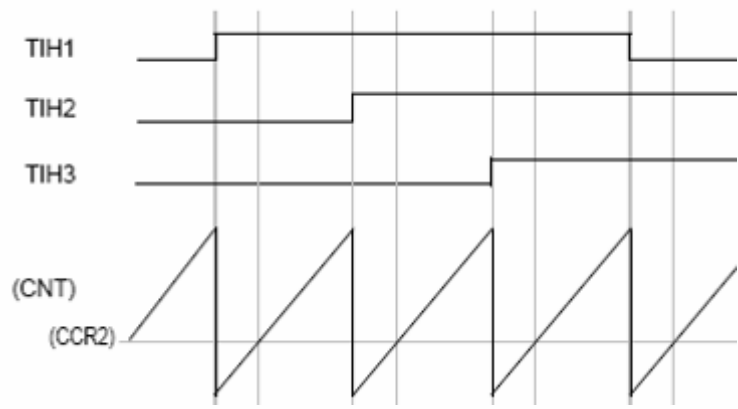
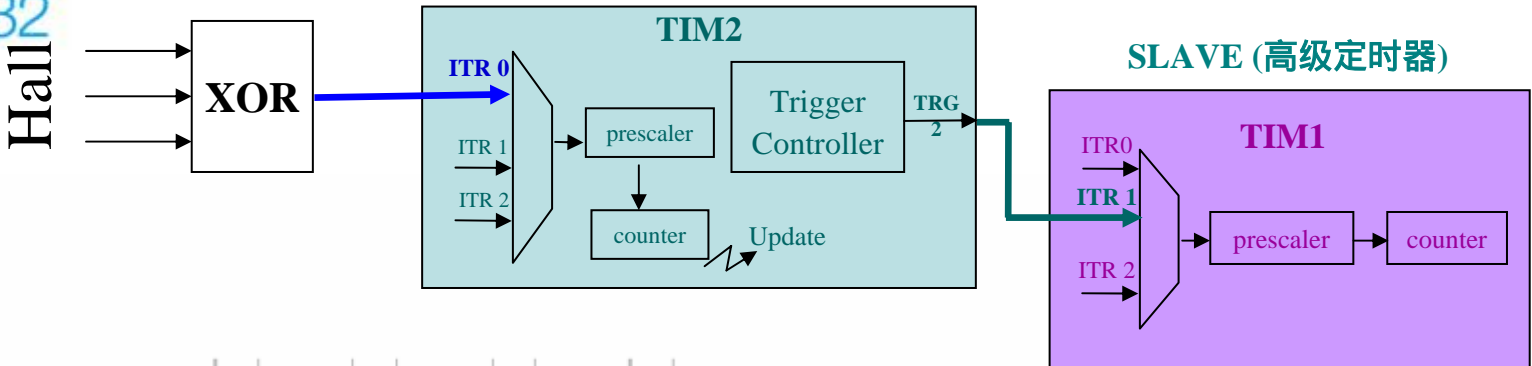
# 例 3/3: 块换相

BLDC马达换相：由一个通用定时器检测Hall位置，触发TIM1定时器换相



MASTER (通用定时器)

SLAVE (高级定时器)





# Plan



## Cortex-M3 简介

-  内核
-  NVIC
-  PWM 产生
-  速度/位置 反馈
-  多定时器配置
-  ADC
-  马达控制套件



# ADC 特性(1/3)



STM32

- ADC转换速度为**1MHZ**，精度为**12位**
  - 采样时间可编程(1.5~239.5cy)，最小采样时间：**107ns**
- ADC输入范围： $0 \leq V_{IN} \leq V_{REF} + (LQFP100 \text{封装才有 } V_{REF} +)$
- 18个通道
  - 16个外部通道
  - 2个内部通道：温度传感器和参考电压
- 中断，包括：
  - 转换结束，注入通道转换结束，ADC看门狗
- DMA
  - 仅ADC1有，但如果ADC2与ADC1同步，可用DMA传输ADC2的转换结果
- 转换通道**编组**
  - 常规转换组：最大16个通道
  - 注入转换组：最大4个通道



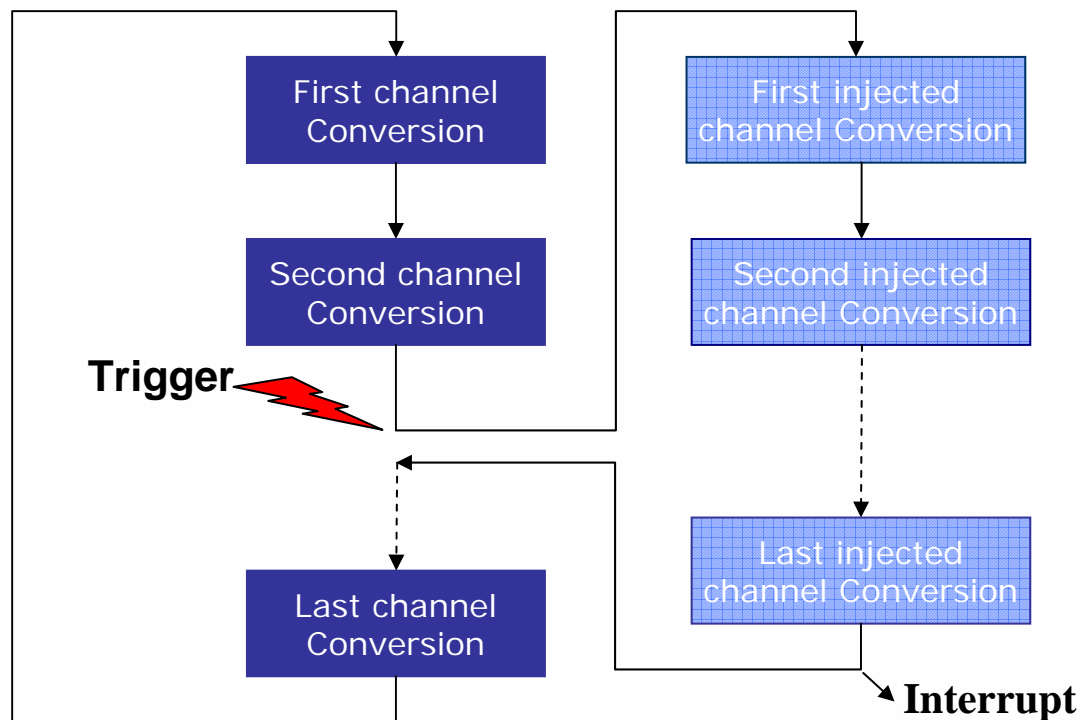
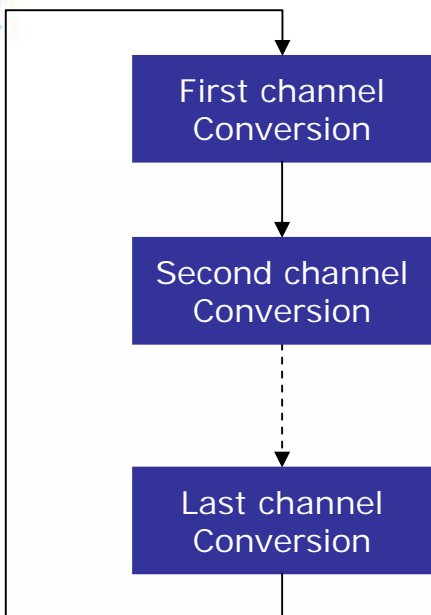
# 注入转换模式



STM32

常规转换扫描模式

注入转换扫描模式



# ADC 特性(2/3)



ADC 看门狗(可针对单个通道，所有常规通道或所有注入通道)

STM32

基于定序器的**扫描模式**：

- 任意通道，任意次序
- 最大16个通道的常规转换（结果由DMA存储）
- 最大4个通道的注入转换（结果相应寄存器存储）

多触发源

- 每个组可被来自定时器的6个事件触发
- 可由外部事件和软件触发



# ADC 特性(3/3)



- 转换数据可向左或向右对齐

STM32

- 4个偏移补偿寄存器

- 补偿外部电路的偏移，如运放。如需要可提供带符号值

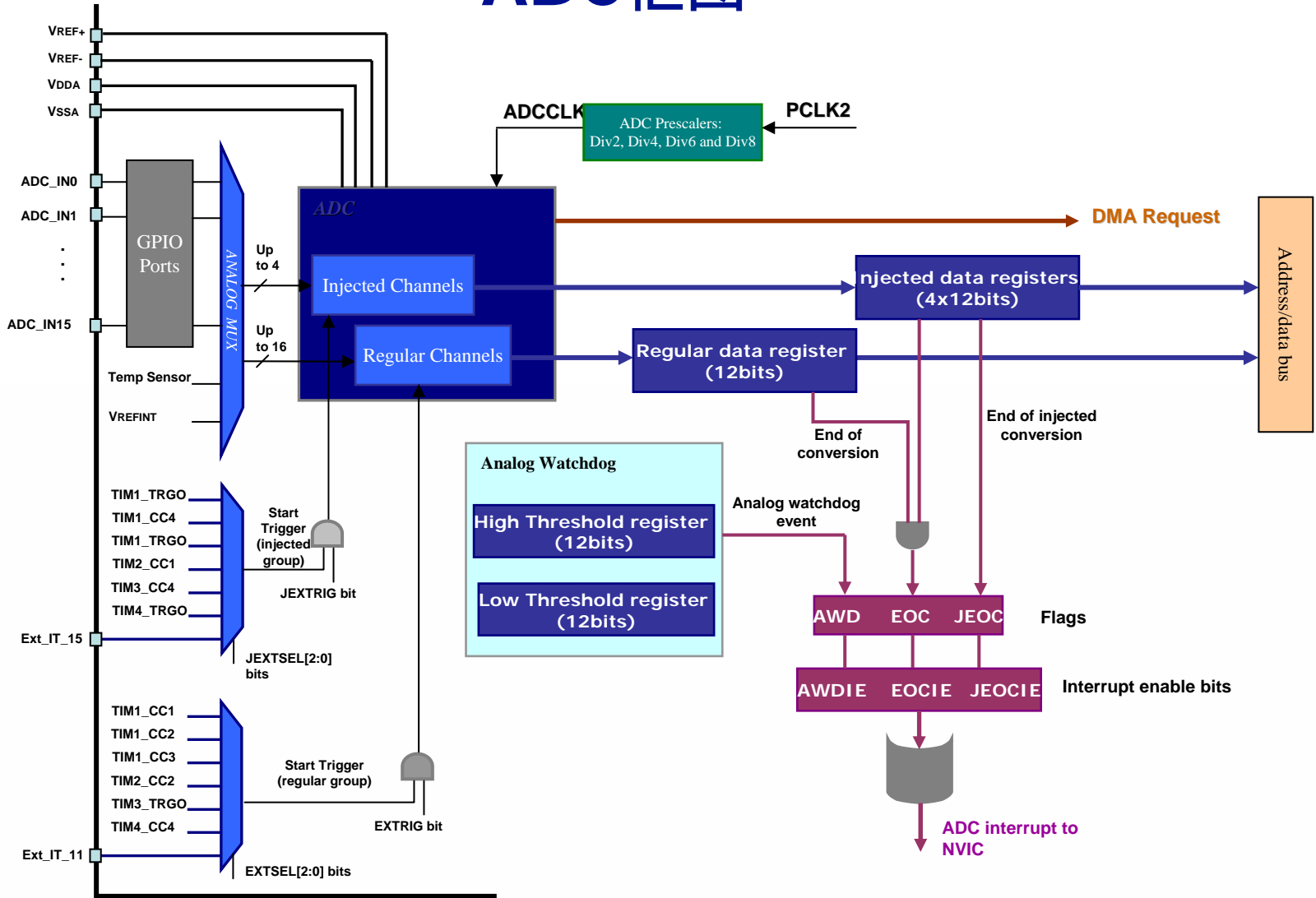
- 每个通道可单独编程采样时间，可以采样不同输入阻抗的信号

- 从1.5cy( $R_{in} < 1.2K$ )到239.5cy( $R_{in} < 350K$ )，共8个值

- 当采样率为1MSps时，可不用电压跟随器



# ADC框图



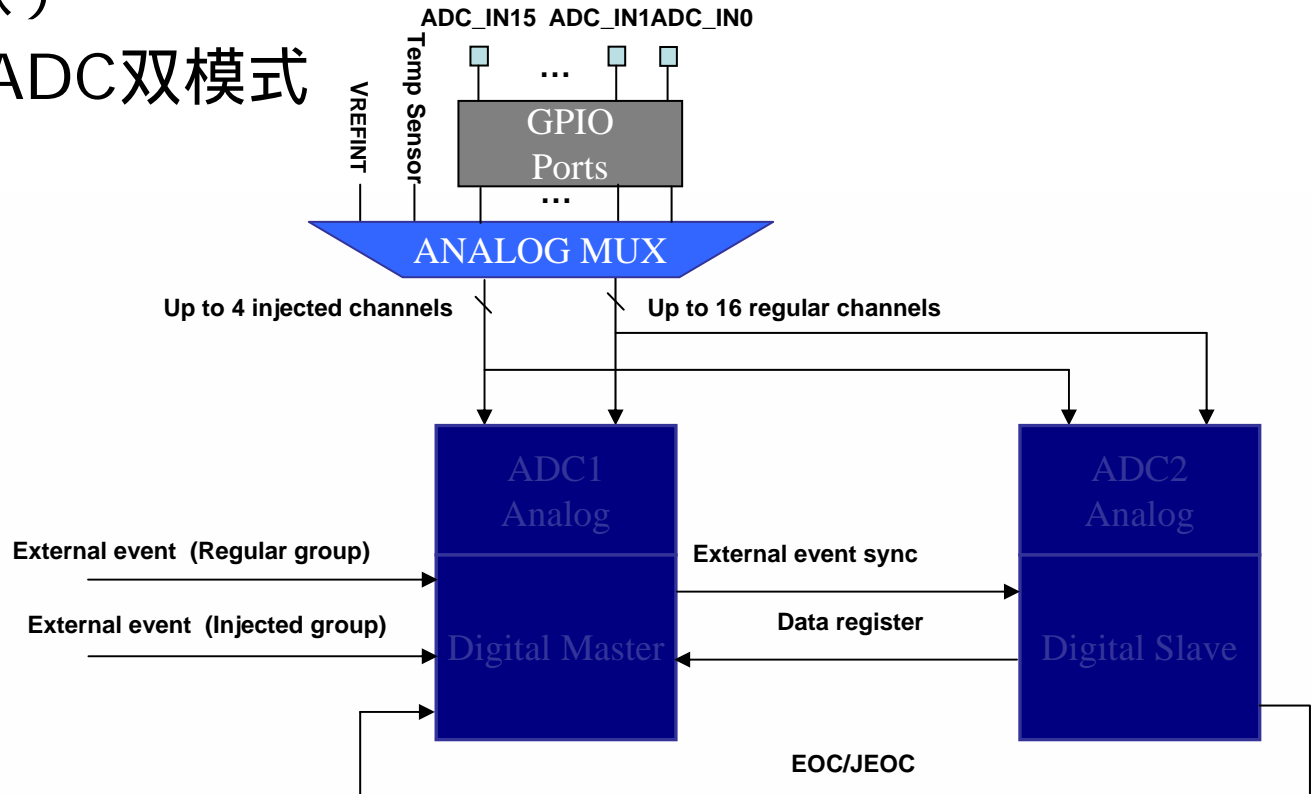
# ADC双模式(1/2)



只能在拥有2个ADC的MCU中实现

ADC1和ADC2可分别单独使用或耦合在一起使用（分主 / 从）

8个ADC双模式

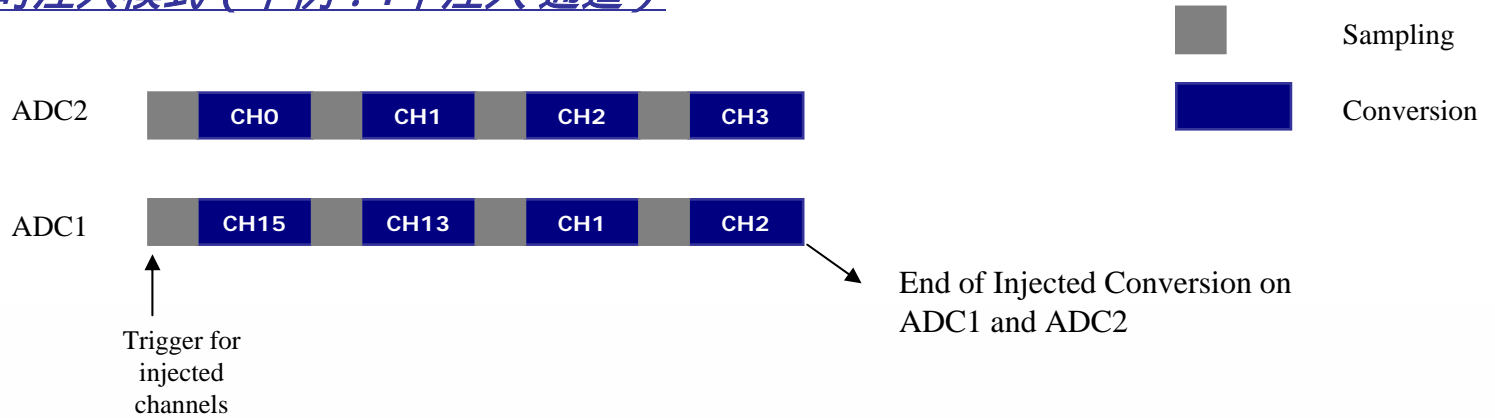


# ADC双模式(2/2)

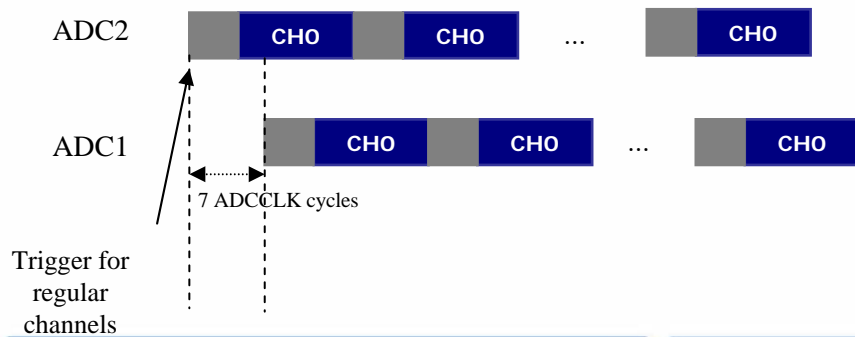


STM32

## 同时注入模式 (下例: 4个注入通道)



## 快速隔行扫描模式: 针对单个常规通道, 且工作在连续模式下



Up to 2 MSps data throughput (DMA-based)



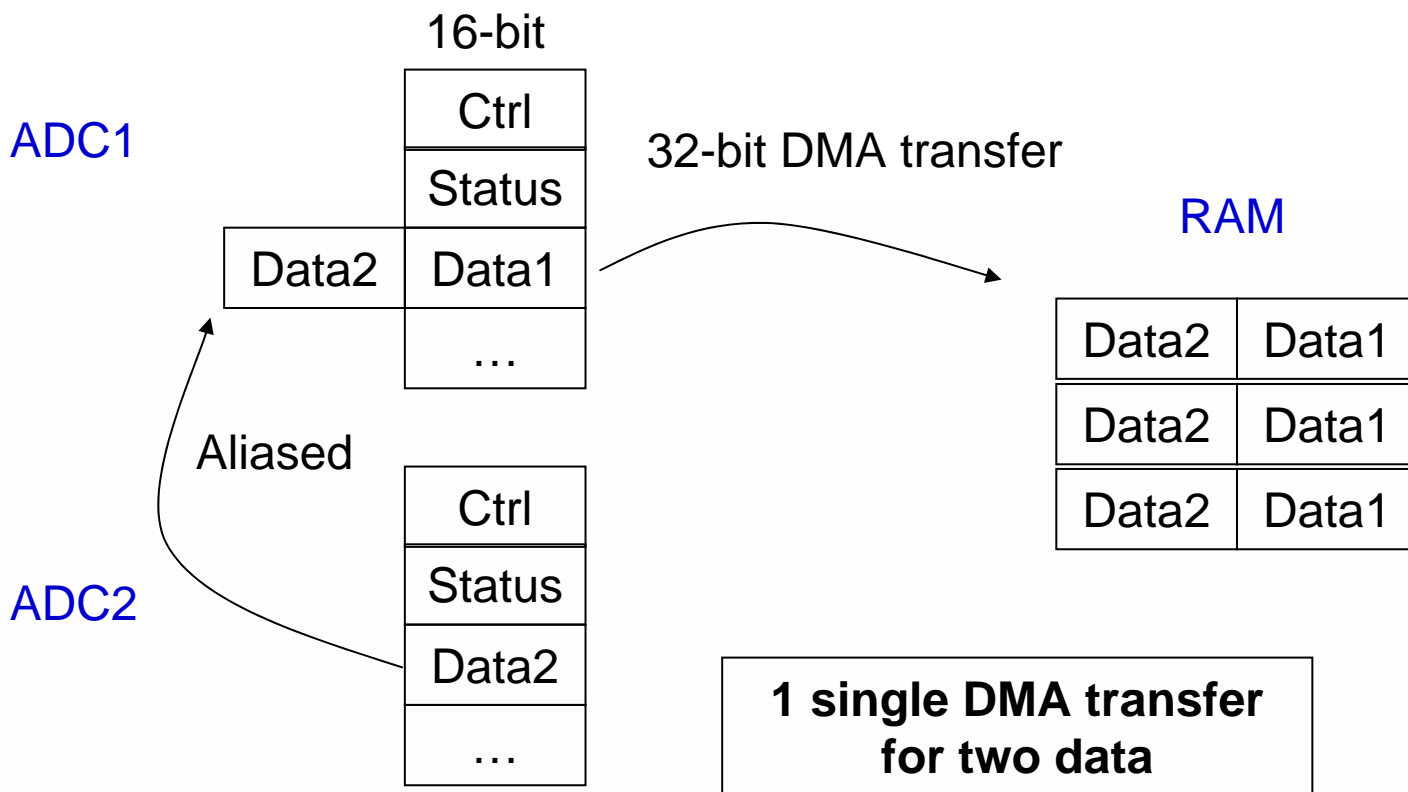


# 隔行扫描模式下的DMA传输



隔行扫描模式：两ADC连续对同一通道采样

STM32



# STM32如何实现电流同时采样



▣ PWM定时器中的同步单元可实现ADC同步

STM32

▣ 可有2个选择：

- ▣ 直接由PWM定时器计数器的峰顶、谷底或两者中的任一个同步
- ▣ 由PWM定时器的第4个输出比较所产生的延时同步

▣ ADC的结果可由“转换完成”中断处理或由DMA存储

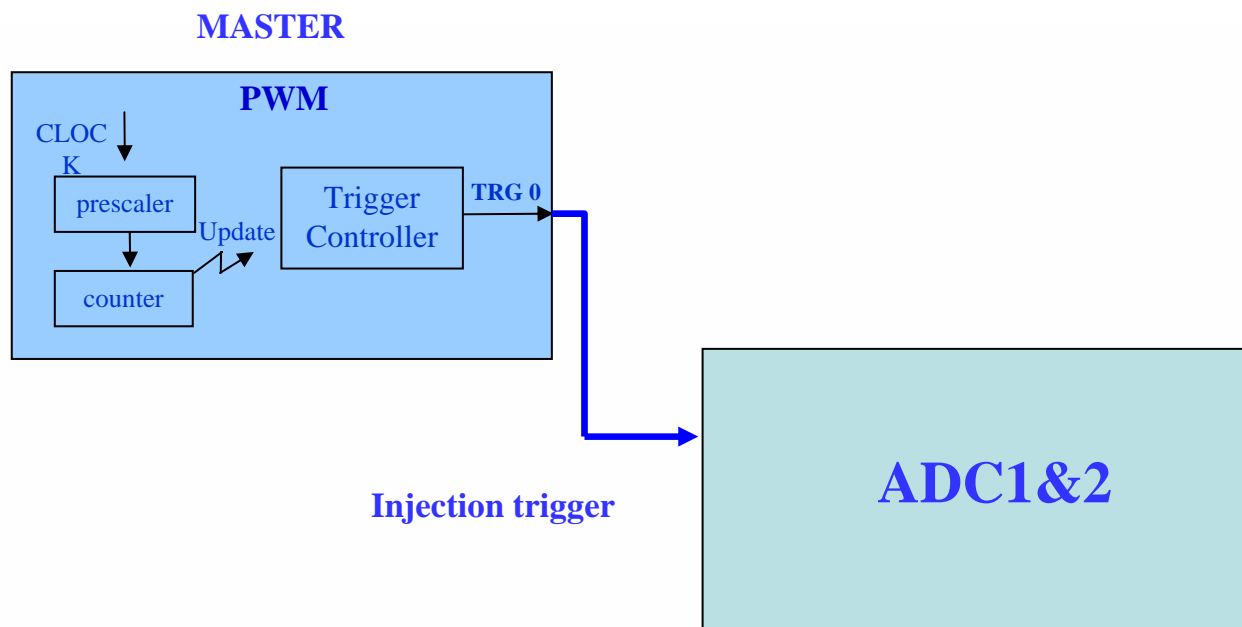


# 直接由U事件进行同时采样



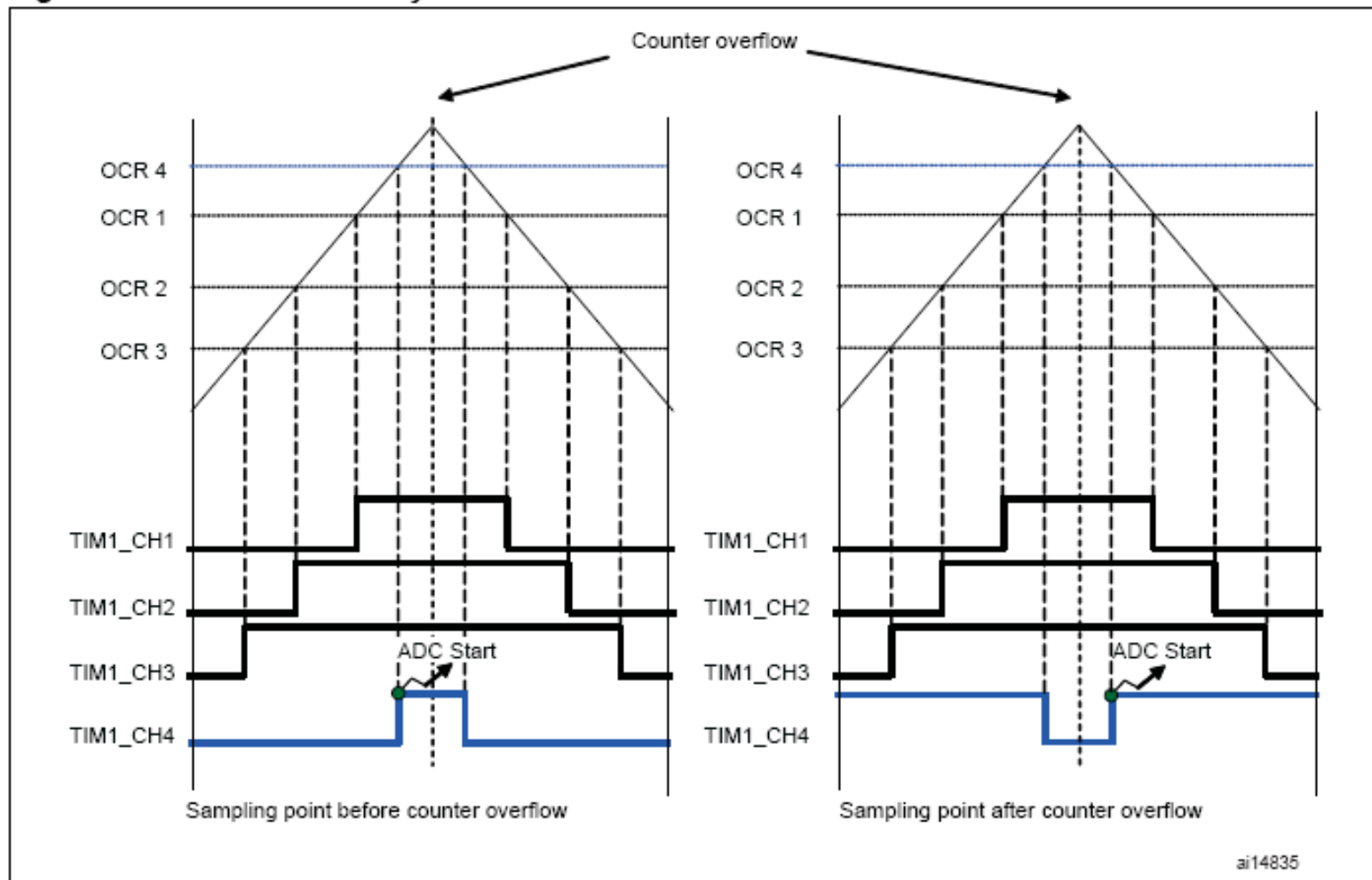
▣ PWM定时器的U事件触发两个ADC，使其工作在“同时注入”转换模式下

▣ 由于连续采样相电流，不会产生误差



# 由TIM1的CH4输出进行同时采样

Figure 34. PWM and ADC synchronization



# Plan



STM32

## ▣ Cortex-M3 简介

▣ 内核

▣ NVIC

▣ PWM 产生

▣ 速度/位置 反馈

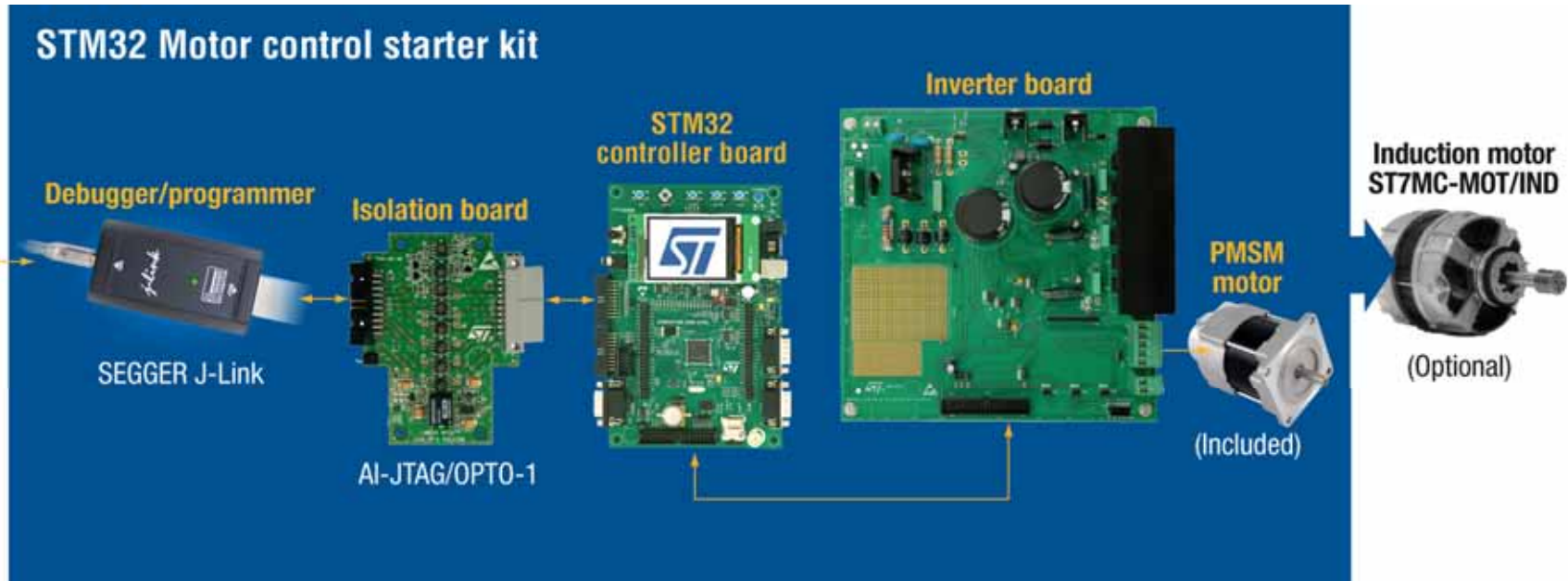
▣ 多定时器配置

▣ ADC

▣ 马达控制套件



# STM32 演示套件



- ❑ 三相半桥功率驱动板，3电阻法读取相电流（单电阻法正在设计中）/电流传感器法
- ❑ 完整的软件库可驱动PMSM或AC马达（AC：带速度传感器）
- ❑ 该套件包含一个带编码器和Hall传感器的PMSM马达